

RECEIVED
CENTRAL FAX CENTER

OCT 27 2006

REMARKS

Reconsideration of this application, as amended, is respectfully requested.

Claims 1-25 are pending. Claims 1-25 have been rejected.

Claims 1, 10, 18, 20, 22, and 24 have been amended. No claims have been canceled. No claims have been added. Support for the amendments is found in the specification, the drawings, and in the claims as originally filed. Applicants submit that the amendments do not add new matter.

Applicants reserve the rights with respect to the applicability of the Doctrine of Equivalents.

REJECTIONS UNDER 35 U.S.C. § 102

Claims 1, 4, 5, 18, 19, 22 and 23 have been rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,076,144 to Peled, et al. ("Peled").

Applicants have amended claim 1 to include defining at least one stream comprising a sequence of basic blocks in the dynamic execution trace, wherein only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch and end the at least one stream on a taken branch, the remaining basic blocks in each stream each ending in a branch instruction, the execution of which does not cause program flow to branch.

Peled discloses identifying of the potential entry points into trace segments. More specifically, Peled discloses

Each cache line stores an instruction trace segment comprising one or more basic blocks of instructions that are predicted to be sequentially executed. For example, in an embodiment where each cache line comprises two basic blocks of instructions, the second basic block of instructions includes instructions to be executed if the branch instruction located at the end of the first basic block is taken.

(Peled, col. 1, lines 22-28) (emphasis added)

Portion of the Peled cited by the Examiner merely discloses that the trace segment has control flow instructions, and the addresses of control flow instructions within the trace segment are used to access the trace segment (Figures 2, 3, col. 4, lines 1-12).

Thus, Peled merely discloses the trace segment that comprises two basic blocks, wherein the second basic block of instructions includes instructions to be executed if the branch instruction located at the end of the first basic block is taken. In contrast, amended claim 1 refers to defining at least one stream comprising a sequence of basic blocks in the dynamic execution trace, wherein only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch, so that the stream ends on a taken branch, the remaining basic blocks in each stream each ending in a branch instruction, the execution of which does not cause program flow to branch.

Because Peled fails to disclose all the limitations of amended claim 1, Applicants respectfully submit that amended claim is not anticipated by Peled under 35 U.S.C. § 102(b).

Given that claims 4, 5, 18, 19, 22 and 23 contain related limitations, Applicants respectfully submit that claims 4, 5, 18, 19, 22 and 23 are not anticipated by Peled under 35 U.S.C. § 102(b).

REJECTIONS UNDER 35 U.S.C. § 103

Claims 2, 3 and 6-9 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Peled, as applied in claims 1 and 5, in view of U.S. Patent No. 5,933,626 to Mahalingaiah ("Mahalingaiah").

Applicants respectfully disagree. It is respectfully submitted that Peled does not teach or suggest a combination with Mahalingaiah, and Mahalingaiah does not teach or suggest a combination with Peled. Peled teaches identifying multiple entry points into trace elements to improve caching of decoded micro-ops. Mahalingaiah, in contrast, teaches saving the state of the

microprocessor to improve the speed of the instruction tracing (col. 1, lines 40-col. 3, line 20). It would be impermissible hindsight, based on Applicants own disclosure, to combine Peled and Mahalingaiah.

Furthermore, even if Peled and Mahalingaiah were combined, such a combination would lack defining at least one stream comprising a sequence of basic blocks in the dynamic execution trace, wherein only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch and end the at least one stream on a taken branch, the remaining basic blocks in each stream each ending in a branch instruction, the execution of which does not cause program flow to branch, as recited in amended claim 1.

Therefore, Applicants respectfully submit that amended claim 1 is not obvious under 35 U.S.C. § 103(a) over Peled in view of Mahalingaiah.

Given that claims 2, 3 and 6-9 depend from amended claim 1, and add additional limitations, Applicants respectfully submit that claims 2, 3 and 6-9 are not obvious under 35 U.S.C. § 103(a) over Peled in view of Mahalingaiah.

Claims 10-14 and 20-25 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Peled, in view of U.S. Patent No. 5,627,994 to Levy, et al. ("Levy").

Amended claim 10 includes analyzing each local trace for streams, each stream comprising a sequence of basic blocks that were sequentially executed, wherein only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch and end the at least one stream on a taken branch, the remaining basic blocks in each stream each ending in a branch instruction, the execution of which does not cause program flow to branch.

As set forth above, Peled fails to disclose such limitations of amended claim 10.

It is respectfully submitted that Peled does not teach or suggest a combination with Levy, and Levy does not teach or suggest a combination with Peled. Peled teaches identifying the entry

points into trace segments improve caching of decoded micro-ops. Levy, in contrast, teaches assigning requests to cache memories to improve hit ratio. It would be impermissible hindsight, based on Applicants own disclosure, to combine Peled and Levy.

Furthermore, even if Peled and Levy were combined, such a combination would lack analyzing each local trace for streams, each stream comprising a sequence of basic blocks that were sequentially executed, wherein only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch and end the at least one stream on a taken branch, the remaining basic blocks in each stream each ending in a branch instruction, the execution of which does not cause program flow to branch, as recited in amended claim 10.

Therefore, Applicants respectfully submit that amended claim 10 is not obvious under 35 U.S.C. § 103(a) over Peled in view of Levy.

Given that claims 11-14 and 20-25 contain related limitations, Applicants respectfully submit that claims 11-14 and 20-25 are not obvious under 35 U.S.C. § 103(a) over Peled in view of Levy.

Claims 15-17 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Peled, in view of Levy as applied in claim 14, and further in view of U.S. Patent No. 6,594,773 B1 to Lisitsa, et al. ("Lisitsa").

Amended claim 10 includes analyzing each local trace for streams, each stream comprising a sequence of basic blocks that were sequentially executed, wherein only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch and end the at least one stream on a taken branch, the remaining basic blocks in each stream each ending in a branch instruction, the execution of which does not cause program flow to branch.

As set forth above, Peled fails to disclose such limitations of amended claim 10.

It is respectfully submitted that Peled does not teach or suggest a combination with Levy and Lisitsa, Levy does not teach or suggest a combination with Peled and Lisitsa, and Lisitsa does not teach or suggest a combination with Peled and Levy. Peled teaches identifying the entry points into trace segments improve caching of decoded micro-ops. Levy, in contrast, teaches assigning requests to cache memories to improve hit ratio. Lisitsa, in contrast to Peled and Levy, teaches using a graph of multiple processing modules to control the streaming media data. It would be impermissible hindsight, based on Applicants own disclosure, to combine Peled, Levy, and Lisitsa.

Furthermore, even if Peled, Levy, and Lisitsa were combined, such a combination would lack analyzing each local trace for streams, each stream comprising a sequence of basic blocks that were sequentially executed, wherein only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch and end the at least one stream on a taken branch, the remaining basic blocks in each stream each ending in a branch instruction, the execution of which does not cause program flow to branch, as recited in amended claim 10.

Therefore, Applicants respectfully submit that amended claim 10 is not obvious under 35 U.S.C. § 103(a) over Peled in view of Levy, and further in view of Lisitsa.

Given that claims 15-17 depend from amended claim 10 and add additional limitations, Applicants respectfully submit that claims 15-17 are not obvious under 35 U.S.C. § 103(a) over Peled in view of Lev, and further in view of Lisitsa.

RECEIVED
CENTRAL FAX CENTER

OCT 27 2006

CONCLUSION

It is respectfully submitted that in view of the amendments and arguments set forth herein, the applicable rejections and objections have been overcome.

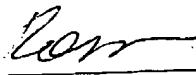
If there are any additional charges, please charge Deposit Account No. 02-2666 for any fee deficiency that may be due.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: October 27, 2006

By: _____


Tatiana Rossin
Reg. No. 56,833

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(408) 720-8300